Chapter 27

Implementing Production Grids for Science and Engineering

William E. Johnston[1], John M. Brooke[2], Randy Butler[3], David Foster[4], Mirco Mazzucato[5]

We examine experience accrued by those who have been involved with building large-scale Grids intended for production use. By this we mean that the primary task of such Grids is to enable large-scale scientific research rather than to act as a test-bed for research into software development, or more speculative computer science. Severe constraints are brought to bear in the construction of such Grids, since access to such resources is subject to stringent requirements of security and high quality of service. We must also reckon with the fact that the resources on such Grids may not be primarily designed or tailored to Grid use. The Grid middleware cannot control policy on such Grids, it must co-operate with the site policy and resource management systems. We may be dealing with ancillary resources, such as telescopes and experimental facilities, that are not computational, may not possess a full operating system, and may have policy and management requirements that are very different from each other and from a computational node. Even within the domain of computational machines, we may encounter multiple architectures, clusters of workstations, specialist massively-parallel machines, clusters of shared-memory nodes, and machines with vector rather than scalar processors. The specialism of such machines means that they may run a variety of operating systems and there may be no common set of software that will work on all of them (for example some architectures may not run a Java Virtual Machine). It is the task of the Grid middleware to combine them in a virtual organization and the task of the Grid building team to solve the particular engineering challenges involved. Even within a Grid that has a relatively homogeneous architecture, the need to provide reliability and high quality of service and availability impose their own constraints. Thus monitoring the Grid becomes a vital task.

We attach great importance to abstracting a set of Core Grid Functions which will have to be implemented in order for the Grid to be productive. Interoperability is a key concept, especially in extending Grids across organizational boundaries. In several of these Grids data is at least of equal importance to computation, thus data-handling and transfer issues must be included in the core functionality. Our structure is as follows: Section 1 summarizes the necessary requirements for Grid middleware in the production environment and the need for the Grid approach. Section 2 motivates our identification of core functions. Section 3 discusses the engineering requirements necessary for establishing security, reliability and quality of service. Importantly, we here describe our experience and discuss its current limitations, taking care to examine the policy-based issues which are crucial to success. In Section 4 we give a schematic walk-through of the steps involved

[1] DOE Lawrence Berkeley National Laboratory and NASA Ames Research Center (USA),wejohnston@lbl.gov

[2] Manchester Computing and Department of Computer Science, University of Manchester (UK), j.m.brooke@man.ac.uk

[3] National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign (USA), rbutler@ncsa.uiuc.edu

[4] CERN LHC Computing Grid Project, david.foster@cern.ch

[5] INFN-Padova (Italy), mirco.mazzucato@pd.infn.it

in establishing a Production Grid and in Section 5 we try to draw some conclusions and look towards the advent of Open Grid Services.

27.1 Introduction: Lessons Learned for Building Large-Scale Grids

Over the past several years there have been a number of projects aimed at building "production" Grids. These Grids are intended to provide identified user communities with a rich, stable, and standard distributed computing environment. By "standard" and "Grids" we specifically mean Grids based on the common practice and standards coming out of the Global Grid Forum (GGF) (www.gridforum.org). By "production" we mean that the Grid has the same levels of reliability and maintainability as more traditional large scale production facilities. The quality of service metrics such as 24x7 operation, mean time between failure (MTBF) rates and throughput should be defined and measurable. What is acceptable for these metrics will depend on the virtual organization (VO) involved, but the infrastructure in terms of user support, engineering support, diagnostic tools, and software release procedures need to be in-place to operate a Grid in a production mode.

There are a number of projects around the world that are in various stages of putting together production Grids that are intended to provide this sort of persistent cyber infrastructure for science. Among these are the UK's e-Science program [1], the European DataGrid [2], NASA's Information Power Grid [3], several Grids under the umbrella of the DOE Science Grid [4], and (at a somewhat earlier sage of development) the Asia-Pacific Grid [5].

In addition to these basic Grid infrastructure projects, there are a number of well advanced projects aimed at providing the sorts of higher-level Grid services that will be used directly by the scientific community. These include, for example, Ninf (A Network based Information Library for Global World-Wide Computing Infrastructure - [6, 7]) and GridLab [8].

This paper, however, addresses the specific and actual experiences gained in building NASA's IPG, DOE's Science Grids, the European Union DataGrid [2], EUROGRID [9], the LHC Computing Grid (LCG) [10], the PACI Alliance Grid [11], and the UK eScience Grid [1], all of which are targeted at infrastructure for large-scale, collaborative science, and access to large-scale computing and storage facilities.

The IPG project at NASA Ames [3] has integrated the operation of Grids into the NASA Advanced Supercomputing (NAS) production supercomputing environment and the computing environments at several other NASA Centers, and together with some NASA "Grand Challenge" application projects, has been identifying and resolving issues that impede application use of Grids.

The DOE Science Grid [4] is implementing a prototype production environment at four DOE Labs and at the DOE Office of Science supercomputer center, NERSC [12]. It is addressing Grid issues for supporting large-scale, international, scientific collaborations.

The LHC Computing Grid is a world wide Grid production Grid project connecting regional centers providing resources for the processing of data coming from the Large Hardron Collider experiment based at CERN in Geneva, Switzerland [13].

The European Union DataGrid has integrated a large number of European sites resources providing the infrastructure for High Energy Physics and, as well, for other sciences like Earth Observation and Biology. This is providing access to large scale data and distributed computing resources to communities of hundreds of scientists distributed all over Europe and belonging to several tens of different national Organizations.

EUROGRID [9] connects together major academic and commercial centers in Europe with emphasis on high-performance applications utilizing specialist architectures, some of which have atypical operating systems and complex architectural configurations.

The UK eScience Grid [1] connects major centers of eScience in the UK, forming the platform for the projects in the UK science program.

The Italian INFN Grid [14] project integrates the INFN resources located in about 20 major Italian Universities and provide a major component of the basic Grid infrastructure aiming at supporting the national e Science activities.

This paper only describes the experience gained from deploying a specific set of software: Globus [15], Condor [16], SRB/MCAT [17], Virtual Data Toolkit [11] PBSPro [10], and a PKI authentication substrate [18-20]. That is, these suites of software have provided the implementation of the Grid functions used in the Grids described here, with the sole exception of EUROGRID which used UNICORE.

27.1.1 The Grid Software

The Globus package provides:

- A clear, strong, and standards based security model
- Modular functions (not an all or nothing approach) providing all of the Core Grid Functions, except general events
- A clear model for maintaining local control of resources that are incorporated into a Grid
- A general design approach that allows a decentralized control and deployment of the software.
- A demonstrated ability to accomplish large-scale Metacomputing (in particular, the SF-Express application in the Gusto testbed – see [21])
- Presence in supercomputing environments
- A clear commitment to open source
- Today, one would also have to add “market share”.

SRB/MCAT (metadata catalogue and storage management) and Condor (job manager) were added because they provided specific, required functionality to the IPG and the LCG.

Key to the High Energy Physics and Astrophysics communities are petascale Virtual Data Grids that meet the data-intensive computational needs of a diverse community of thousands of scientists spread across the globe. The concept of Virtual Data encompasses the definition and delivery to a large community of a (potentially unlimited) virtual space of data products derived from experimental data or from simulations. In this virtual data space, requests may be satisfied via direct access and/or by (re)computation of simulation data on-demand, with local and global resource management, policy, and security constraints determining the strategy used. What is stored in the associated metadata of the data products is not necessarily just descriptions of the data and pointers to that data, but prescriptions for generating the data. The Virtual Data Toolkit [11] is providing tools for file naming and location transparency (replica management), and for instantiating data on-demand from metadata descriptions.

Grid software beyond that provided by these suites are being defined by many organizations, most of which are involved in the GGF. Implementations are becoming available, and are being experimented with in the Grids described here (e.g. the Grid monitoring and event framework of the Grid Monitoring Architecture Working Group [22] [22] and the Data Grid Resource Broker [2]), and some of these projects will be mentioned in this paper. Nevertheless the software of the prototype-production Grids described in this paper is provided primarily by the aforementioned packages, and these provide the context of this discussion.

This paper recounts some of the lessons learned in the process of deploying these Grids, and provides an outline of the steps that have proven useful and usually necessary in order to deploy these sorts of Grids. This reflects the work of a substantial number of people, representatives of whom are acknowledged below. The lessons fall into four general areas – deploying operational infrastructure (what has to be managed operationally to make Grids work), establishing cross site trust, dealing with Grid technology scaling issues, and listening to the users – and all of these will be discussed.

This paper is addressed to those that are setting up science oriented Grids, or who are considering doing so.

27.1.2 The Case for Grids

It is difficult at this point to justify quantitatively a business case for Grids in the science environment , however, several qualitative arguments can be provided based on current experience.

The new opportunity offered by a compute Grid is to flexibly re-purpose resources to solve problems in a time critical manner. If a Grid is composed of only a few sites, and each site has a computing resource that is already 100% utilized across many applications, there is little gain in participating in a Grid under normal conditions. However, if the policies are such that for specific problems the whole resource can be used for a period of time then the Grid provides a new capability, allowing solutions that could not be obtained by the single sites. If the Grid involves many sites, with different peak usage time, then there is a gain in distributing work loads over machines that will inevitably be idle for part of the time. So flexibility and efficiency in the use of existing resources are potential Grid benefits .

Data Grids give the possibility of connecting computational resource with data sources in a very flexible way, facilitating, for example, bringing data to the cpu or the job to the data. This is important today because instruments such as particle accelerators, earth observing satellites, telescopes of many varieties, all generate in specific locations highly specialized data that need typically to be made available to large distributed scientific communities. Since computing and storage resource are also normally distributed they are almost never co-located with the data to be analyzed.

Finally, the current big success of Grids is that they are providing the security and directory infrastructure to support the ever increasing size and complexity of science collaborations. The uniform security infrastructure of Globus and the emerging cross-site trust agreements for identity authorities and resource usage that result from the use of Globus are having a major positive impact on large-scale scientific collaboration.

All of these factors relate to the “business case” of Grids and science, but it is hard to quantify the value of new capabilities until the impact (e.g. a more effective and capable process of science) is realized and evaluated.

27.2 The Grid Context

We address here the needs of Grids where productive work is the major driver and Grids that must operate within the operational and security constraints of large pre-existing institutions responsible for complex and valuable resources. This is accomplished through three aspects: 1) A set of uniform core software services that manage and provide access to heterogeneous, distributed resources, 2) a widely deployed infrastructure, and 3) higher level services like the Data Grid tools. The software architecture of such a Grid is depicted in Figure 1.

In the opinion of the authors, there is a set of basic functions that all Grids must have in order to be called a Grid: The Core Grid Functions. These constitute the "neck of the hourglass" of Grids, and in the Grids described here are provided primarily by the Globus software [15]. These core functions include a Grid Information Service ("GIS" – the basic resource discovery mechanism) [23], a Grid Security Infrastructure ("GSI" – the tools and libraries that provide Grid security) [24], a Grid job initiator mechanism (e.g., Globus GRAM [25]), a Grid scheduling function, and a basic data management mechanism such as GridFTP [26]. To complete this set we need a Grid event mechanism. The Grid Forum's Grid Monitor Architecture [27] addresses one approach to Grid events, and there are several prototype implementations of the GMA (e.g. [28, 29] [30] [31]). A communications abstraction (e.g., Globus I/O [32]) that incorporates Grid security is also in this set.

At the resource management level the capabilities of the individual computing system, data system or instrument are exposed and may contain specific functionality necessary for Grid computing. For example, job management systems (e.g., PBSPro [10], Maui [33]) that support advance reservation of resource functions (e.g. CPU sets) are needed to support co-scheduling of administratively independent systems. It is the job of the Grid Information Service to publish this sort of resource capability information so that only systems with this capability would be selected for tasks that require co-scheduling.

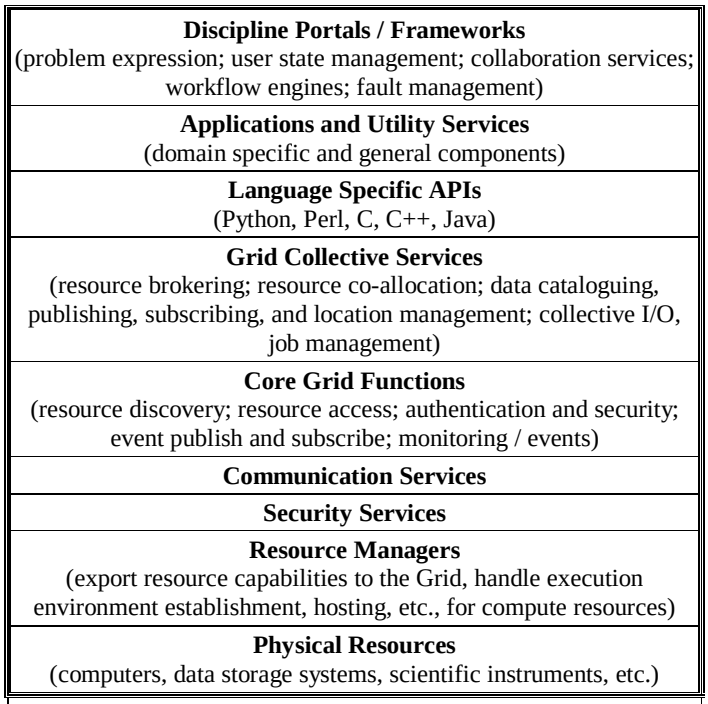

Figure 1. Grid Architecture

Beyond this basic set of common services are associated client-side libraries and tools, and other high level capabilities such as matching job requirements to available resources, for example the EU DataGrid resource broker [2], Condor-G [34] for job management, SRB/MCAT [17] for federating and cataloguing tertiary data storage systems, and the new Data Grid [15, 35] tools for Grid data management.

In this paper, while we focus on the issues of building a Grid through deploying and managing the Core Grid Functions (provided mostly by Globus), we point out along the way other software suites that may be required for the functionality required for specific communities or types of applications, and we also discuss some of the production issues of these other suites.

27.3 Issues for Production Grids

Each type of production environment is characterized by its own key issues that must be addressed for success. Routing agreements, site service level agreements, etc., for networks, user services, accounting and allocation management, system tuning, etc., for supercomputer centers, and so forth.

In the case of Grids there are also key operational concerns that must be carefully addressed, if not to ensure success, at least to make sure that success is not inhibited.

These issues include, for example, the model for the Grid Information Services, cross-site trust, understanding how and why Grids are scoped, management of local authorization, etc.

These issues are discussed in this section.

27.3.1 Grid Information Service (GIS)

The GIS provides capabilities for locating resources based on the characteristics needed by a job (OS, CPU count, memory, etc.). The Globus Monitoring and Discovery Service (MDS) [23] is an information service implementation with two components. The Grid Resource Information Service (GRIS) runs on the Grid resources (computing and data systems) and handles the soft-state registration of the resource characteristics. The Grid Information Index Server (GIIS) is a user accessible directory server that supports searching for resource by characteristics. Other information may also be stored in the GIIS, and the Grid Information Services group of the GGF is defining the schema for various objects [36]. Currently a common schema definition and implementation by the Globus and EU DataGrid projects have been achieved for Grid Computing Element and Storage Element description.

The basic functions of the GIS, in terms of Globus MDS-2, are illustrated in Figure 2. These consist of resource identity, registration target(s), and the registration process. Registration both conveys the resource information to the GIS and has a soft-state maintenance mechanism (the "keep-alives" in the figure).

A GIIS should be planned at each distinct site with significant Grid accessible resources. This is important in order to avoid single points of failure, because if you depend on a GIIS at some other site, and it becomes un-available, you will not be able to examine your local resources. Depending upon the number of local resources, it may be necessary to set up several GIISs at a site in order to accommodate the search load.

An initial GIS model could be to use independent GIISs at each site. In this configuration, either cross-site searches require explicit knowledge of each of the GIISs that have to be searched independently, or all resources cross-register in each GIIS. (Where a resource registers is a configuration parameter in the GRISs that run on each Grid resource.)

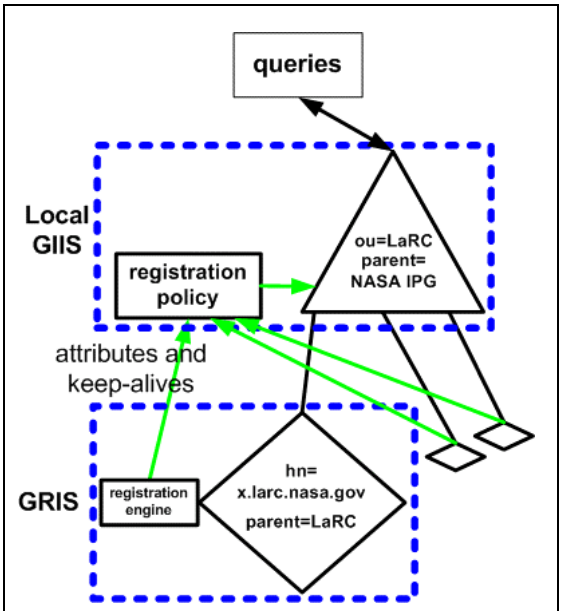

Figure 2. Basic GIS Functions

(The lines indicate relationships, the arrows information flow, the diamonds resources, and the triangles are the information directory.)

It is important to consider early on what kind of information needs to be managed by the GIS. Most of the information available today is resource specific, documenting such things as the operating systems, hardware configuration, and more. This information is useful in understanding the capabilities of the resources at a high level and may and is likely to be input to higher-level resource brokers that filter all possible candidates against other user criteria. Users of the Grid will likely have a list of information they would want to find within the GIS. In some cases it might include, e.g., data storage availability, software versions, or it may be pointers to other information sources for information such as the status of a specific job. All of this is important to consider when deciding what information to store in the GIS. It is also very important to follow the developing schema definition standards to ensure Grid applications interoperability.

The Model for the Grid Information System

Directory servers above the local GIISs are an important scaling mechanism for several reasons. An organization may have multiple GIISs at the "local" level for administrative, security management, or load balancing and there needs to be a single point of entry of users. The higher-level GIISs [37] expand the resource search space through automated cross-GIIS searches for resources, and therefore provide a potentially large collection of resources transparently to users. They also provide the potential for query optimization and query results caching. Further more, such directory services provide the possibility for hosting and/or defining virtual organizations (by storing this non-resource information in the same GIS data hierarchy as the resource characteristics), and for providing federated views of collections of data objects that reside in different storage systems.

There are currently two main approaches that are being used for building directory services above the local GIISs. One is a hierarchically structured set of directory servers and a managed namespace usually based on X.500[6] style naming. The other is a set of "index" servers that provide ad-hoc, or virtual organization specific, views of a specific set of other servers such as a collection of GIISs, data collections, etc., not unlike a project Web page that provides access to all of the information relevant to that project.

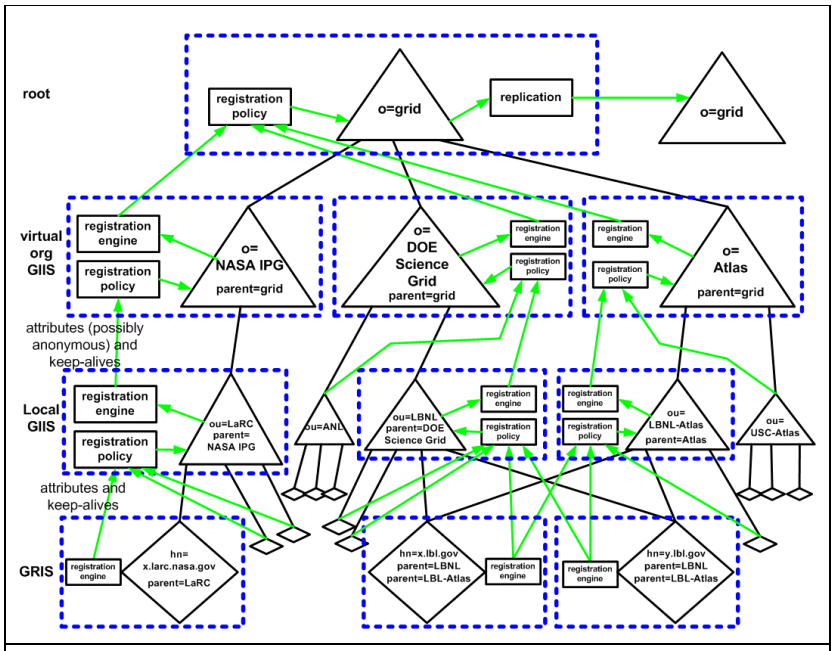

Figure 3. The GIS Can Encode Organizational Relationships.

Both provide for "scoping" your Grid in terms of the resource search space. The structuring of these higher level GIISs may provide scoping for a VO or site, or may reflect a general structure for an actual organization. In principle, there can be a fair bit of complexity to this structure, and except for the root (no formal root exists today), the situation depicted in Figure 3 is not atypical. In particular, individual resources may register with however many (potentially unrelated) GIISs are appropriate for their function.

An X.500 Style Hierarchical Name Component Space Directory Structure

Using an X.500 Style hierarchical name component space directory structure has the advantage of organizationally meaningful names that represent a set of "natural" boundaries for scoping searches and it also means that you can potentially use commercial meta-directory servers for better scaling. An issue to consider here is the ability to keep the hierarchy up to date if mapped onto a physical organizational structure that may be subject to frequent change.

Attaching virtual organization roots, data name spaces, etc., to the hierarchy makes them automatically visible, searchable and in some sense "permanent" (because they are part of this managed name space).

---

[6] ISO OSI Directory Standard (CCITT Recommendation X.500). See, e.g., [38].

If you plan to use this approach, try very hard to involve someone who has some X.500 experience because the directory structures are notoriously hard to get right, a situation that is compounded if VOs are included in the namespace.

Index Server Directory Structure

The Globus MDS ([23]) information system implementation has added to the usual LDAP based directory service capabilities several features that are important for Grids.

Soft-state registration provides for auto registration and de-registration of resources as well as registration access control. It keeps the information up-to-date (via a keep-alive mechanism) and it provides for a self configuring and dynamic Grid: A new resource registering for the first time is essentially no different than an old resource that is re-registering after a system crash for example. The auto-registration mechanism also allows resources to participate in multiple information hierarchies thereby easily accommodating membership in multiple VOs. The registration mechanism also provides a natural way to impose authorization on those who would register with your GIISs.

Every directory server from the GRIS on the resource, up to and including the root of the information hierarchy, is essentially the same, which simplifies the management of the servers.

Other characteristics of MDS include:

- resources are typically named using the components of their DNS name which has the advantage of using an established and managed name space
- one must use separate “index” servers to define different relationships among GIISs, virtual organization, data collections, etc., on the other hand, this allows you to establish “arbitrary” relationships within the collection of indexed objects
- hierarchical GIISs (index nodes) are emerging as the preferred approach in the Grids community that uses the Globus software.

Apart from the fact that all of the directory servers must be run as persistent services and their configuration maintained, the only real issue with this approach is that we do not have a lot of experience with scaling this to multiple hierarchies with thousands of resources.

27.3.2 Cross-Site Trust Management

The process of authorizing user access to Grid resources implies several major important steps:

- Physical identification of the user identity by the Registration and Certification Authorities
- Attribution of roles and general authorization by Virtual Organization (VO) management
- Grant of local authorization by the local resource owner and security managers
- And, in some cases, inclusion of the user into the Grid VO LDAP directory

One of the most important contributions of Grids to supporting large-scale collaboration is the uniform Grid entity naming and authentication mechanisms provided by the Grid Security Infrastructure.

However, for this mechanism to be useful, the collaborating sites and organizations must establish mutual trust in the authentication process most Grids use the software mechanism of PKI, X.509 identity certificates, and their use in the GSI through TLS/SSL ([39]). These mechanisms are understood and largely accepted.

PKI uses both a public and private key to prove – or authenticate – that user's identity. Public keys are well known and readily available on Web sites and at other public locations. Private keys are

known only by the particular user and are encrypted. The public key is associated with a named identity in an identity certificate, which is used for Grid authentication.

In order to obtain an identity certificate, a user submits the public key half of their private/public key pair, along with the user identity (naming), and a request for a digital certificate, to a third party, called a Certificate Authority (CA). The CA verifies the identity of the user, the validity of the requested name (which may include organization name elements – e.g. o=Lawrence Berkeley National Lab, ou=Computing Sciences, cn=William E. Johnston – to ensure that the user has not requested a name that makes it look like it is a member of an organization for which the particular CA is not the naming authority), and binds the user's identity to the public key by digitally signing a document – the identity certificate – that contains the public key, the user name, and the identity of the certification authority. This certificate, together with a challenge from the resource that requires the use of the user's private key to satisfy, is the authentication process: the user's proof of identity. If, for example, the user wants to logon to a remote compute resource, he or she presents this digital certificate and demonstrates possession of the private key as proof of identity. The user interaction is all on the local system – no passwords are sent to the remote machines. This is accomplished using the cryptographic protocols of TLS. (However, in most cases the user does not login to remote systems, but rather create a proxy certificate that authenticates to the remote resource on behalf of the user when, e.g., a batch job is submitted. This proxy mechanism provides a "single sign-on" for the use. Having authenticated once on the local systems, the proxy mechanism securely carries that authentication to remote systems. See [24].)

The real issue is that of establishing trust in all of the various processes involved in issuing and using the certificate. This process varies somewhat depending on the trust model of the CA and/or the VOs that use the CA. However, typically the trust issues include:

- The process that each Certification Authority ("CA") uses for issuing the identity certificates to users and other entities, such as host systems and services. This involves two steps.
  - First is the "physical" identification of the entities, and a verification of their association with a Virtual Organization. This usually involves a step where the VO authorizes the CA to issue a certificate, once user identity is established.
  - The second is the process by which an X.509 certificate is issued. Both of these steps are defined in the CA policy.
- The use of the certificate once it is issued, and the management of the user's private key.

These issues are described in more detail, below.

In the PKI authentication environment assumed here, the CA policies are encoded as formal documents associated with the operation of the Certification Authority that issues Grid identity credentials. These documents are called the Certificate Policy / Certification Practice Statement, and we will use "CP" to refer to them collectively. (See [40].)

Trust

Trust is "confidence in, or reliance on, some quality or attribute of a person or thing, or the truth of a statement."[7] Cyberspace trust starts with clear, transparent, negotiated, and documented policies associated with identity. When a Grid identity token (X.509 certificate in the current context) is presented for remote authentication and is verified using the appropriate cryptographic techniques, then the relying party should have some level of confidence that the person or entity that initiated the transaction is the person or entity that it is expected to be.

[7] Oxford English Dictionary, Second Edition (1989). Oxford University Press.

The nature of the policy associated with identity certificates depends a great deal on the nature of particular Grid communities and/or the Virtual Organizations associated with those Grids. It is relatively easy to establish policy for homogeneous communities, such as in a single organization, because an agreed upon trust model will likely already exist. It is difficult to establish trust for large, heterogeneous virtual organizations involving people from multiple, international institutions, because the shared trust models do not exist. The typical issues related to establishing trust may be summarized as:

- Across administratively similar systems
  - e.g., within an organization
  - informal / existing trust model can be extended to Grid authentication and authorization
- Administratively diverse systems
  - e.g., across many similar organizations (e.g. NASA Centers, DOE Labs)
  - formal / existing trust model can be extended to Grid authentication and authorization
- Administratively heterogeneous
  - e.g., cross multiple organizational types (e.g. science labs and industry)
  - e.g., international collaborations
  - formal / new trust model for Grid authentication and authorization will need to be developed

The process of getting a CA's CP (and therefore the Grid user's certificates) accepted by other Grids (or even by multi-site resources within a Grid) involves identifying the people who can authorize remote users at all of the sites and organizations that form a VO / collaboration, and exchanging CPs with them. The CPs are evaluated by each party in order to ensure that local policy for remote user access is met. If it is not, then a period of negotiation ensues. The sorts of issues involved are indicated in the European Union DataGrid Acceptance and Feature matrices ([41]).

The policy in the CA CP represents a trust model, and there is not a single trust model for VOs and CAs. The trust model that gives rise to much of what is described here is a model that is appropriate to the sort of mission-oriented organizations and VOs that are typical in large-scale science like high energy physics, or at government operated laboratories. We probably do not have the experience yet to catalogue trust models, but it is clear that others will show up in Grids. University oriented VOs or projects may very well find the procedures of the mission-oriented VOs too elaborate and/or restrictive. Government agencies or corporations that need to protect proprietary information may find them insufficient.

It substantially simplifies the process if the sites of interest already have people who are 1) familiar with the PKI CP process, and 2) focused on the scientific community of the institution rather than on the administrative community. Further, it is vital to get the site security people involved at an early stage in the planning and implementation so that process is appropriately institutionalized.

Cross-site trust relationships may, or may not, be published. Frequently it is. See, e.g., the European Union DataGrid list of acceptable CAs ([42]).

It should be re-emphasized at this point that gaining access to resources is a two step process: First the user has to authenticate (prove identity) and second that identity then must be authorized to use the resource (e.g. by virtue of having an allocation on the resource). We are discussing the authentication step here.

Establishing an Operational CA[8]

Set up, or identify, a Certification Authority to issue Grid X.509 identity certificates to users and hosts. IPG, NCSA [43] the DOE Science Grids, and many European CAs, such as INFN, use the Netscape CMS software ([44]) for their operational CA because it is a mature product that allows a very scalable usage model that matches well with the needs of science Virtual Organizations.

People setting up Grids must take care to understand the issues associated with the CP of their CA. As noted, one thing governed by CP is the "nature" of identity verification needed to issue a certificate, and this is a primary factor in determining who will be willing to accept a CA's certificates as adequate authentication for resource access. Changing this aspect of the CP could well mean not just re-issuing all certificates, but requiring all users to re-apply for certificates.

CPs are highly stylized documents, and a standard template should be used. The GGF is working on a standard set of CPs that can be used as templates, and the DOE Science Grid has developed a CP that supports international collaborations, and that is contributing to the evolution of the GGF CP. (The DOE Science Grid CP is at http://www.doegrids.org/ [40] and the GGF CP may be found at http://www.gridforum.org/2_SEC/GCP.htm)

CPs have to account for all of the various entities for which the CA will have to issue certificates. These typically include human users, hosts (systems), services (e.g. GridFTP), and possibly security domain gateways (e.g. the PKI to Kerberos gateway, KX509 [45]). Each of these must have a clear policy and procedure described in the CA's CP/CPS.

A Grid CP established and published as early as possible in the process of setting up a Grid so that issues are exposed as soon as possible. Also, when setting up Grids / VOs that will have to interoperate with other CAs, the discussions about homogenizing the CPs should begin as soon as possible, as this can be a lengthy process.

It should be noted that certificates are, as described in the CP, issued for particular purposes. Certificates issued by Grid CAs are generally only valid for authentication. Other potential uses for certificates are digital signature by the user of email, documents, etc., and for general encryption. Support for digital signatures and encryption have additional requirements that likely are not satisfied by Grid authentication processes, and the Grid CA's CP may explicitly disallow it's certificates use for these purposes.

Naming

One of the important issues in developing a CP is the naming of the principals (the "subject," i.e. the Grid entity identified by the certificate). While there is an almost universal tendency to try and pack a lot of information into the subject name (which is a multi-component, X.500 style name), increasingly there is an understanding that the less information of any kind put into a certificate, the better. This simplifies certificate management and re-issuance when users forget passphrases (which will happen with some frequency). More importantly, it emphasizes that ***all trust is local*** – that is, established by the resource owners and/or when joining a virtual community. The main reason for having a complicated subject name invariably turns out to be that people want to do some of the authorization based on the components of the name (e.g. organization). However, this usually leads to two problems. One is that people belong to multiple organizations, and the other is that the authorization implied by the issuing of a certificate will almost certainly collide with some aspect the authorization actually required at any given resource. It is also the case that the projects, organizations and affiliations that people are associated with may change frequently even though

[8] Much of the work described in this section is that of Tony Genovese (tony@es.net) and Mike Helm (helm@es.net), Energy Sciences Network, Lawrence Berkeley National Laboratory.

they remain members of a VO, and revoking and reissuing authentication certificates just to change organizational aspects of the names is not an efficient way to handle those changes.

The CA run by ESnet (the DOE Office of Science scientific networks organization [46]) for the DOE Science Grid and the NCSA Alliance CA, for example, serve several dozen different Virtual Organizations, several of which are international in their makeup. The certificates use what is essentially a flat namespace, with a "reasonable" common name (e.g. a "formal" human name) to which has been added a random string of alphanumeric digits to ensure name uniqueness.

However, if you do choose to use hierarchical institutional names in certificates, don't use colloquial names for institutions – consider their full organizational hierarchy in defining the naming hierarchy. Find out if anyone else in your institution, agency, university, etc., is working on PKI (most likely in the administrative or business units) and make sure that your names do not conflict with theirs, and if possible follow the same name hierarchy conventions.

It should be pointed out that CAs set up by the business units of scientific organizations frequently do not have the right policies to accommodate Grid users. This is not surprising since they are typically aimed at the management of institutional financial transactions.

The Certification Authority Model

There are several models for CAs, however increasingly associated groups of collaborations / Virtual Organizations are opting to fund a single CA provider.

The primary reason for this is that for the trust model being described here (mission-oriented VOs, as described above) it is a formal and expensive process to operate a CA in such a way that it will be trusted by others, and multiple groups are opting to share the cost of a CA.

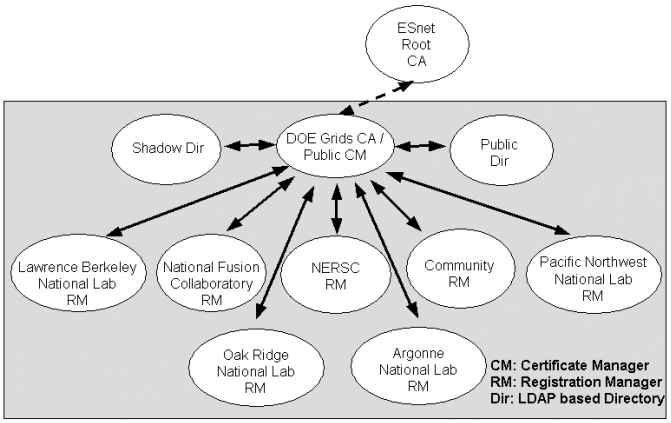

Figure 4. Software Architecture for 5/15/02 Deployment of the DOE Grids CA.

(Courtesy Tony Genovese (tony@es.net) and Mike Helm (helm@es.net), ESnet, Lawrence Berkeley National Laboratory.)

One such model has a central CA that has an overall CP, and has subordinate policies for each of the VOs that it serves. The CA delegates to VOs (via Registration Agents) the responsibility of deciding who is a member of the particular VO, and how the subscriber / user will be identified in order to be issued a VO certificate. Each VO has an appendix in the CP that describes VO specific issues. VO Registration Agents are responsible for applying the CP identity policy to their users and other entities. Once satisfied that a user meets the requirements, the RA authorizes the CA to issue (generate and sign) a certificate for the subscriber.

This is the model of the DOE Science Grid CA, for example, and it is intended to provide a CA that is scalable to dozens of Virtual Organizations and thousands of users. This approach to scalability is the usual divide and conquer, together with a hierarchical organization that maintains the policy integrity. The architecture of the DOE Science Grid CA is indicated in Figure 4, and has the following key features.

The Root CA (which is kept locked up and off-line) signs the certificates of the CA that issues user certificates. With the exception of the "community" Registration Manager, all RMs are operated by

the VOs that they represent. (The community RM addresses those "miscellaneous" people who legitimately need DOE Grid certificates, but for some reason are not associated with a Virtual Organization.) The process of issuing a certificate to a user ("subscriber") is indicated in Figure 5.

ESnet operates the CA infrastructure for DOE Science Grids, they do not interact with users. The VO RAs interface with certificate requestors. The overall policy oversight is provided by a Policy Management Authority, which is a committee that is chaired by ESnet, and is comprised of each RA, and a few others.

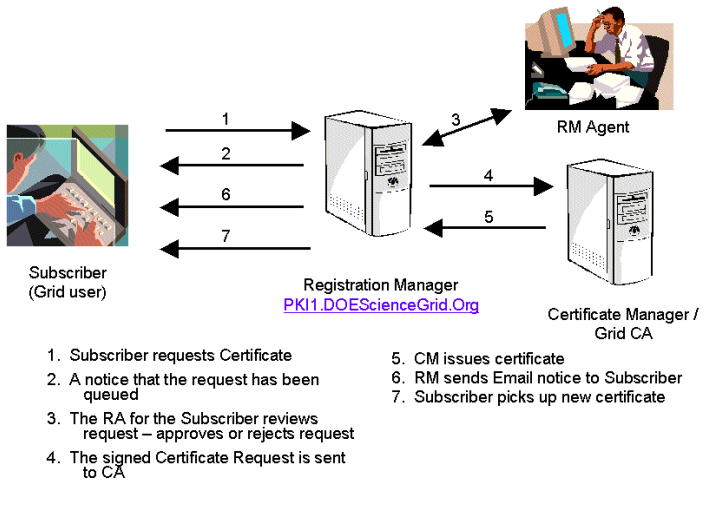

Figure 5. Certificate Issuing Process

(Courtesy Tony Genovese (tony@es.net) and Mike Helm (helm@es.net), ESnet, Lawrence Berkeley National Laboratory.

This approach uses an existing organization (ESnet) that is set up to run a secure, production infrastructure (its network management operation) to operate and protect the critical components of the CA. ESnet defers user contact to agents within the collaboration communities. In this case, the DOE Science Grid was fortunate in that ESnet personnel were also well versed in the issues of PKI and X.509 certificates, and so they were able to take a lead role in developing the Grid CA architecture and policy

27.3.3 Defining / Understanding the Extent of "Your" Grid

The "boundaries" of a Grid are primarily determined by three factors:

o Interoperability of the Grid software

Many Grid sites run some variation of the Globus software, and there has been fairly good interoperability between versions of Globus, so most Globus sites can potentially interoperate.

A basic level of interoperability involving simple jobs to single sites has also been established between Globus and UNICORE, in particular transfer of UNICORE to Globus certificates (and vice versa) has been established. Translation and transfer of more complex requests is being worked on in [47].

Interoperability between the virtual data toolkit (VDT) Grid suite and the European DataGrid technology has also been demonstrated at a basic level. The interoperability issues are being worked on as part of the Grid Laboratory Universal Environment (GLUE) [48] both at a practical level with today's tools and specifying common schemas that will facilitate interoperability.

o What CAs the resources of your Grid trust

This is explicitly configured on each Globus resource on a per CA basis.

Your trusted CAs establish the maximum extent of your user population, however there is no guarantee that every resource in what you think is "your" Grid trusts the same set of CAs – i.e. each resource potentially has a different space of users – this is a local decision. In

fact, this will be the norm if the resources are involved in multiple virtual organizations as they frequently are, e.g. in the high energy physics experiment data analysis communities.

- How you scope the searching of the GIS/GIISs or control the information that is published in them

  This depends on the model that you choose for structuring your directory services.

So, the apparent “boundaries” of most Grids depend on who is answering the question.

27.3.4 Local Control and Authorization

As of yet there is no standard authorization mechanism for Grids. Almost all current Grid software uses some form of access control lists (“ACL”), which is straightforward, but typically does not scale very well.

The Globus mapfile is an ACL that maps from Grid identities (the subject names in the identity certificates) to local UIDs on the systems where jobs are to be run. The Globus Gatekeeper ([25]) replaces the usual login authorization mechanism for Grid based access and uses the mapfile to authorize access to resources after authentication. Therefore, managing the contents of the mapfile is the basic Globus user authorization mechanism for the local resource.

The mapfile mechanism is good in that it provides a clear-cut way for locally controlling access to a system by Grid users. However, it is bad in that for a large number of resources, especially if they all have slightly different authorization policies, it can be difficult to manage.

The first step in the mapfile management process is usually to establish a connection between user account generation on individual platforms and requests for Globus access on those systems. That is, generating mapfile entries is done automatically when the Grid user goes through the account request process. If your Grid users are to be automatically given accounts on a lot of different systems with the same usage policy it may make sense to centrally manage the mapfile and periodically distribute it to all systems. However, unless the systems are administratively homogeneous, a non-intrusive mechanism, such as an email to the responsible system admins to modify the mapfile, is best. The NCSA Alliance developed a transaction based account management system that supports account management and Grid mapfile management in a virtual organization that spans administrative domains. This system is now being extended through a NSF NMI [49] grant.

The Globus mapfile also allows a many-to-one mapping so that, for example, a whole group of Grid users can be mapped to a single account. Whether or not the individual identity is preserved for accounting purposes is typically dependent on whether the batch queuing system can pass the Grid identity (which is carried along with a job request, regardless of the mapfile mapping) back to the accounting system. PBSPro, e.g., will provide this capability (see section 0).

One way to address the issues of mapfile management and disaggregated accounting within an administrative realm is to use the Community Authorization Service (CAS), which is just now being tested. See [50].

Finally, the administrators of a resource can apply a secondary check on authorization by tailoring the CA naming policy to ensure that on certificates with an appropriate name structure are validated. This is a blanket check applied to all certificates.

27.3.5 Site Security Issues

Incorporating any computing resource into a distributed application system via Grid services involves using a whole collection of IP communication ports that are otherwise not used. If your

systems are behind a firewall, then these ports are almost certainly blocked and you will have to negotiate with the site security folks to open the required ports.

Globus can be configured to use a restricted range of ports, but it still needs several tens or so, (depending on the level of usage of the resources behind the firewall), in the mid 700 range. A Globus "port catalogue" is available to tell what each Globus port is used for, and this will provide information that site security staff will likely want to know. It will also allow estimating how many ports have to be opened (how many per process, per resource, etc.). Additionally, GIS/GIIS needs some ports open and the CA typically uses a secure Web interface (port 443). The Globus port requirements are given in [51]. This is a document that provides the basis of defining a Grid firewall policy document that hopefully will serve the same role as the CA Certificate Practices Statement: It will lay out the conditions for establishing trust between the Grid administrators and the site security staff who are responsible for maintaining firewalls for site cyber protection.

It is important to develop tools/procedures to periodically check that the ports remain open. Unless you have a very clear understanding with the network security staff, the Grid ports will be closed by the first network engineer that looks at the router configuration files and has not been told why these non-standards ports are open. This is another example of why the various administrators of the systems and networks need to embrace the Grid and see it as part of their responsibility to support it.

Alternate approaches to firewalls have various sorts of service proxies manage the intra service component communication so that one, or no, new ports are used. One interesting version of this approach that was developed for Globus 1.1.2 by Yoshio Tanaka at the Electrotechnical Laboratory (ETL, which is now the National Institute of Advanced Industrial Science and Technology (AIST)) in Tsukuba, Japan, is documented in [52] and [53].

27.3.6 High Performance Communications Issues

If you anticipate high data-rate distributed applications, whether for large-scale data movement or process-to-process communication, then enlist the help of a WAN networking specialist and check and refine the network bandwidth end-to-end using large packet size test data streams. (Lots of problems that can affect distributed application do not show up by pinging with the typical 32 byte packets.) Problems are likely between application an host and site LAN/WAN gateways, WAN/WAN gateways, and along any path that traverses the commodity Internet.

The DOE Science Grid has developed tools and techniques for detecting and correcting these sorts of problems, both in the areas of diagnostics and tuning [54]. NSF funds the NLANR team [55], which also provides tools, support, and documentation on network monitoring and problem resolution.

End-to-end monitoring libraries/toolkits (e.g. NetLogger [56]), Iperf [57], and pipechar [58]) are invaluable for application-level distributed debugging. Iperf was developed as a modern alternative to older tools like ttcp for measuring TCP and UDP bandwidth performance. Iperf is a tool to measure maximum TCP bandwidth, allowing the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss. NetLogger provides for detailed data path analysis, top-to-bottom (application to NIC) and end-to-end (across the entire network path) and is used extensively in the DOE Grid for this purpose. It is also being incorporated into some of the Globus tools. (For some dramatic examples of the use of NetLogger to debug performance problem in distributed applications see [59], [60], [61], and [62].)

If at all possible, provide network monitors capable of monitoring specific TCP flows and returning that information to the application for the purposes of performance debugging. (See, e.g., [63]).

In addition to identifying problems in network and system hardware and configurations, there are a whole set of issue relating to how current TCP algorithms work, and how they must be tuned in order to achieve high performance in high-speed, wide area networks. Increasingly, techniques for automatic or semi-automatic setting of various TCP parameters based on monitored network characteristics, are being used to relieve the user of having to deal with this complex area of network tuning that is critically important for high performance distributed applications. See, e.g., [64], [65], and [66].

27.3.7 Batch Schedulers[9]

There are several functions that are important to Grids that Grid middleware cannot emulate and these functions must be provided by the resources themselves.

Some of the most important of these are the functions associated with job initiation and management on the remote computing resources. As part of the NASA IPG project several important features were added to PBS [10] in order to support Grids, in particular advance reservation to support co-scheduling, and we describe several of these features here by way of example.

In addition to the scheduler providing a good interface for Globus GRAM/RLS (which PBS did), one of the things that was found was that people can become quite attached to the specific syntax of the scheduling system. In order to accommodate this PBS was componentized and its user interfaces and client-side process manager functions were packaged separately and interfaced to Globus for job submission. This enables PBS-managed jobs to be run on Globus-managed systems, as well as the reverse. This lets users familiar with PBS to use the PBS front-end utilities (submit via PBS 'qsub' command-line and 'xpbs' GUI, monitor via PBS 'qstat', and control via PBS 'qdel', etc.) to run jobs on remote systems managed by Globus.

This approach is also supported in Condor-G, which, in effect, provides a Condor interface to Globus.

PBS can provide time-of-day based advanced reservation. It actually a queue that "owns" the reservation, and as such, all the access control features (allowing/disallowing specific users/groups) can be used to control access to the reservation. It also allows one to submit a string of jobs to be run during the reservation. The existing job-chaining features in PBS may be used to do complex operations like: run X; if X fails, run Y; if X succeeds, run Z.

PBS passes the Grid user ID back to the accounting system. This is important for allowing, e.g., the possibility of mapping all Grid users to a single account (and thereby not having to create actual user accounts for Grid user) but at the same time still maintaining individual accountability, typically for allocation management.

Finally, PBS supports access-controlled, high priority queues. This is of interest in scenarios where a project might have to "commandeer" a lot of resources in a hurry to address a specific, potentially emergency, situation. Consider, for example, that there is a collection of Grid machines that have been designated for disaster response / management. For such use to be accomplished transparently, we need both lots of Grid managed resources, and ones that have high priority queues that are accessible to a small number of pre-approved people who can submit "emergency" jobs. For immediate response this means that they would need to be pre-authorized for the use of these queues, and that PBS has to do per queue, UID based access control. Further, these should be pre-emptive high-priority queues. That is, when a job shows up in the queue, it forces other,

---

[9] Thanks to Bill Nitzberg (bill@computer.org), one of the PBS developers and Area co-Director for the GGF Scheduling and Resource Management Area, for contributing to this section.

running, jobs to be checkpointed, rolled out, and/or killed, in order to make sure that the high priority job runs.

PBS has full “pre-emption” capabilities, and that, combined with the existing access control mechanisms, provides this sort of “disaster response” scheduling capability. There is a configurable “preemption threshold’ – if a queue's priority is higher than the preemption threshold, then any jobs ready to run in that queue will preempt all running work on the system with lower priority. This means that multiple levels of preemption are possible. The preemption action can be configured to: a) Try to checkpoint, b) Suspend, and/or c) Kill and requeue, in any order.

For access control, every queue in PBS has an access control list that can include and exclude specific users and groups.

27.3.8 MyProxy Service

A frequent Grid user complaint relates to the constant management of GSI credentials, and the frequent necessity of generating proxy credentials so that Grid work can proceed. A related, and functionally more serious issue, is that in order to minimize the risk of the relatively unprotected proxy credentials their lifetimes are kept relatively short (typically 12 hours). This can create significant problems when, e.g., large jobs take longer than that to execute on remote computing systems, or if the batch queues are long, and the proxies expire before execution starts. In either case the job is likely to fail.

The MyProxy service ([67] [68]) is designed to alleviate these problems, as well as to ease the problem of trying to move the user’s permanent identity credential to all of the systems from which the user will want to access the Grid.

The MyProxy service provides for creating and storing intermediate lifetime proxies that may be accessed by, e.g., Web based portals, job schedulers, etc., on behalf of the user. There are plans to extend the service so that it can manage the user’s permanent identity credential as well.

MyProxy provides a set of client tools that let the user create, store, and destroy proxies, and for programs acting on behalf of the user to obtain valid (short term) proxies. The user can create a proxy with a lifetime of a week, or a few weeks, then store that proxy on a MyProxy server. The user and the MyProxy service establish a shared secret, and the user passes that secret to processes that need to obtain proxies on the user’s behalf. In this way, a Grid service such as the Globus job initiator or the Condor job manager, can, after getting the user’s access secret for MyProxy, contact the MyProxy service each time that they need a short term proxy to perform a task. Now when a Grid job manager finds that a job’s proxy is about to expire, it can ask the MyProxy service for a new proxy without user intervention.

The user still has to supply proxy generating authority to MyProxy, but much less often than to a usual Grid task.

The security risks of this are analyzed in [67], and are found to be not only acceptable compared with direct user management of the short-lived proxies, but perhaps even less risky since the process is much less user error prone.

A key operational issue is that not only does the MyProxy server have to be managed as a persistent service, but as a secure persistent service. This means that it should probably be in a controlled physical environment (e.g. a controlled access machine room), should be a strictly single purpose system, and should probably be behind a content filtering firewall.

27.4 Building a Multi-site, Computational and Data Grid

27.4.1 The Grid Building Team

Like networks, successful Grids involve almost as much sociology as technology, and therefore establishing good working relationships among all of the people involved is essential.

The concept of an Engineering Working Group ("WG") has proven successful as a mechanism for promoting cooperation and mutual technical support among those who will build and manage the Grid. The WG involves the Grid deployment teams at each site and meets weekly via teleconference. This WG should include individuals that represent the resources (system, network and security administrators), applications developers, and most importantly those that can set or interpret site policies. Interactions between organizations are bounded by the policies at each site that are often difficult to decipher. There should be a designated WG lead responsible for the agenda and managing the discussions. If at all possible, involve Grid technology experts at least during the first several months while people are coming up to speed. There should also be a WG mail list that is archived and indexed by thread. Notes from the WG meetings should be mailed to the list. This, then, provides a living archive of technical issues and the state of your Grid. These meetings must be held on a regular basis and should be driven by a well-defined set of goals, milestones, and responsibilities.

Grid software involves not only root owned processes on all of the resources, but also a trust model for authorizing users that is not typical. Local control of resources is maintained but is managed a bit differently from current practice. It is therefore very important to set up liaisons with the system administrators for all systems that will provide computation and storage resources for your Grid. This is true whether or not these systems are within your organization.

In many organizations the system, network and security administrators are not well integrated with the Grid deployment teams, this is a critical flaw. These administrators must buy into the Grid concept and see the deployment and support of the Grid as a high priority. Configurations on these systems change frequently and it is easy enough to overwrite or otherwise negate Grid services that were previously deployed through system upgrades. In addition the whole concept of a Grid may directly conflict with an organizations security profile and the security team must embrace the goals of the Grid and work proactively to meet both the needs dictated by site security and the Grid.

27.4.2 Grid Resources

As early as possible in the process, identify the computing and storage resources to be incorporated into your Grid. In doing this be sensitive to the fact that opening up systems to Grid users may turn lightly or moderately loaded systems into heavily loaded systems. Batch schedulers may have to be installed on systems that previously did not use them in order to manage the increased load.

When choosing a batch scheduler, carefully consider the issue of co-scheduling! Many potential Grid applications need this, e.g., to use multiple Grid systems to run cross system MPI jobs or support pipelined applications across systems. Only a few available schedulers currently provide the advance reservation mechanism that is used for Grid co-scheduling (e.g. PBSPro [10] and Maui [33]). If you plan to use some other scheduler be very careful to critically investigate any claims of supporting co-scheduling to make sure that they actually apply to heterogeneous Grid systems. (Several schedulers support co-scheduling only among schedulers of the same type and/or within administratively homogeneous domains.) See the discussion of the PBS, above.

In addition the Grid pushes the operational requirements of the resources so that they may support usage on an extended operational window. Remote access may include the need to support the

usage of these resources across time zones and indeed may require that the resources be operationally support 24/7.

27.4.3 Initial Testbed – First Steps

Use PKI authentication and initially use certificates from the Globus Certificate Authority ("CA"), or most any other CA that will issue you certificates for this test environment. (The OpenSSL CA [69] may be used for this testing. Also see [70].) Issue certificates for the Globus hosts and the testers. These are "toy" certificates that will just be used to get an initial configuration up and running.

Install Globus, the host certificates, and the CA naming policy files on at least two different systems. Populate the mapfile When this works, install, or gain access to, Globus at two or three different sites, preferably more. If you have site firewalls, this is good time to look at the Globus port catalogue [51] and to start talking with whomever controls the firewall policy.

Validate access to, and operation of, the GIS/GIISs at all sites, and test local and remote job submission.

27.4.4 Transition to a Prototype-Production Grid – First Steps

Set up, or identify, your production CA as described previously.

Issue host certificates for all the computing and data resources and establish procedures for installing them and their associated CA naming policy files. Issue user certificates.

Count on revoking and re-issuing all of the certificates at least once before going operational. This is inevitable if you have not previously operated a CA.

Work out and test Grid mapfile and certificate revocation procedures.

Using certificates issued by the Grid CA, validate correct operation of the Grid Security Infrastructure (GSI) [24], GSS libraries, GSI-SSH [71], and GSI-FTP [72] and/or GridFTP [72] [26] at all sites. Install one of the testing monitors that periodically tests each critical service and reports the result in a Web page so that it is easily accessible. (For example, see [73] and [74].)

Start training a Grid application support team on this prototype.

27.4.5 Preparing for Users

Try and find problems before your users do. Design test and validation suites that exercise your Grid in the same way that applications are likely to use your Grid.

As early as possible in the construction of your Grid, identify some test case distributed applications that require reasonable bandwidth, and run them across as many widely separated systems in your Grid as possible, and then run these test cases every time something changes in your configuration.

Establish user help mechanisms, including a Grid user email list and a trouble ticket system. Provide user oriented Web pages with pointers to documentation, including a Globus "Quick Start Guide" [75] that is modified to be specific to your Grid, and with examples that will work in your environment (starting with a Grid "hello world" example).

27.4.6 Moving from Testbed to Prototype Production Grid

At this point Globus, the GIS/MDS, and the security infrastructure should all be operational on the testbed system(s). The Globus deployment team should be familiar with the install and operation issues, and the system admins of the target resources should be engaged.

Deploy and build Globus on at least two production computing platforms at two different sites. Establish the relationship between Globus job submission and the local batch schedulers (one queue, several queues, a Globus queue, etc.)

Validate operation of this configuration.

27.4.7 Grid Systems Administration Tools[10]

Grids present special challenges for system administration due to the administratively heterogeneous nature of the underlying resources.

SDSC is developing a Grid configuration tool called *gridconfig*. Gridconfig will present a single unified interface to all your Grid software including today, Globus, Condor, and NWS. Gridconfig will be released as part of the NMI Release 2 [49].

In the DOE Science Grid monitoring tools are built as pyGlobus ([76] [77]) modules for the NetSaint [78] system monitoring framework. These currently allow testing GSIftp, MDS and the Globus gatekeeper. There are plans for a GUI tool that will use these modules to allow an administrator to quickly test functionality of a particular host.

The NCSA Grid in a Box effort [73] has developed a simple Grid service monitor that has proven to be very effective. Status of the Grid services are tested on a regular basis and can be readily verified.

The harder issues in Grid administrator tools revolve around authorization and privilege management across site boundaries. So far the work has been concentrated only tools for identifying problems. When problems are located, the current way to correct them is to email a privileged local user on the broken machine in order to fix things. Longer term a framework that will use a more autonomic model for continuous monitoring and restart of services needs to be developed.

In several Grids, tools and techniques are being developed for extending Trouble Ticket based problem tracking systems to the Grid environment.

In the future Grid account systems must evolve to track Grid user usage across a large number of machines and manages allocations in accordance with (probably varying) policy on the different systems. Some work by Jarosław Nabrzyski and his colleagues at the Poznan Supercomputing and Networking Center [79] in Poland is developing prototypes in this area. See [80].

27.4.8 Data Management and Your Grid Service Model

Establish the model for moving data between ***all*** of the systems involved in your Grid.

GridFTP servers should be deployed on the Grid computing platforms and on the Grid data storage platforms.

This presents special difficulties when data resides on user systems that are not usually Grid resources, and raises the general issue of your Grid "service model:" What services are necessary to support in order to achieve a Grid that is useful for applications, but are outside of your core Grid

---

[10] Thanks to Keith Jackson (krjackson@lbl.gov) and Stephen Chan (sychan@lbl.gov) for contributing to this section.

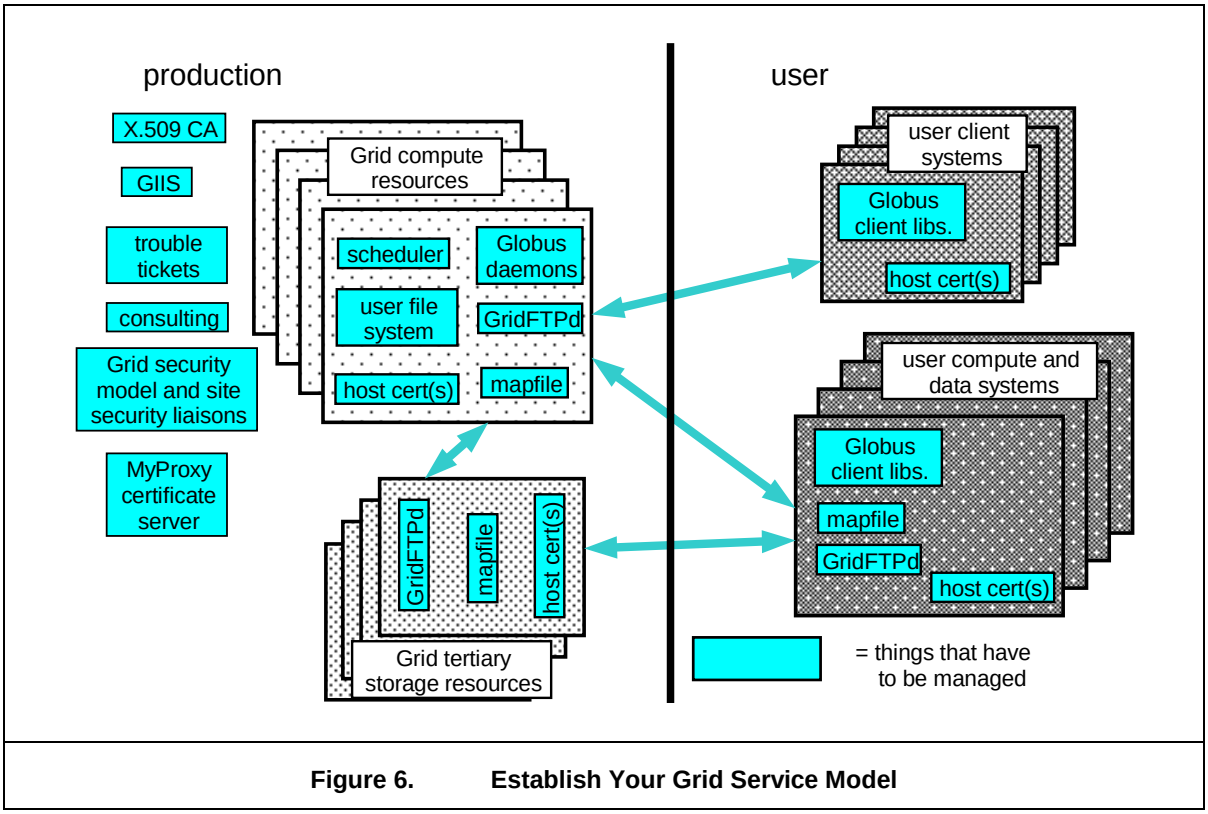

Figure 6. Establish Your Grid Service Model

resources (e.g. GridFTP on user data systems), and how you will support these services, are issues that have to be recognized and addressed.

Determine if any user systems will manage user data that are to be used in Grid jobs. This is common in the scientific environment where individual groups will manage their experiment data, e.g., on their own systems. If user systems will manage data, then the GridFTP server should be installed on those systems so that data may be moved from user system to user job on the computing platform, and back.

Offering GridFTP on user systems may be essential, however managing long lived / root access Grid components on user systems may be "tricky" and/or require you to provide some level of system admin on user systems.

Validate that all of the data paths work correctly.

These issues are summarized in Figure 6.

27.4.9 Take Good Care of the Users as Early as Possible

If at all possible, establish a Grid/Globus application specialist group. This group should be running sample jobs as soon as the testbed is stable, and certainly as soon as the prototype-production system is operational. They should be able to assist generally with building Grid distributed applications, and specifically should serve as the interface between users and the Grid system administrators in order to solve Grid related application problems.

Identify specific early users and have the Grid application specialists encourage / assist them in getting jobs running on the Grid

One of the scaling / impediment-to-use issues currently is that extant Grid functions are relatively primitive (i.e., at a low level). This is being addressed by Grid middleware at various levels that provide aggregate functionality, more conveniently packaged functionality, toolkits for building

Grid based portals, etc. Examples of such work in progress includes the Web Grid Services (e.g. the Open Grid Services Architecture [81] and the resulting Open Grid Services Interface [82] work at GGF), the Grid Web services testbed of the GGF GCE Working Group [83]), diverse interfaces to Grid functions (e.g., PyGlobus [77], CoG Kits [84], [85], and [86]), and the Grid Portal Development Kit [87]).

One approach that has been successful in the IPG and DOE Science Grid is to encourage applications that already have their own "frameworks" to port those frameworks on the Grid. This is typically not too difficult because many of these frameworks already have some form of resource management built in, and this is easily replaced / augmented with Grid resource management functions. This hides some of the "low level functionality" problem. Examples of such frameworks deployed in IPG and/or DOE Science Grid are NPSS [88, 89] and Cactus [11]. Another example of this approach is Ninf [6].

Another useful tool for users is a Grid job tracking and monitoring portal such as IPG's LaunchPad [90] and NPACIs HotPage [91].

27.5 Conclusions

We have presented the essence of experience gained in building several production Grids, and provided some of the global context for this work.

As the reader might imagine, there were a lot of false starts, refinements to the approaches and to the software, and several substantial integration projects (e.g. the NASA sponsored SRB and Condor integration with Globus) to get where we are today.

However, the point of this paper is to try and make it substantially easier for others build production Grids. This is what is needed in order to move us toward the vision of a common cyber infrastructure for science.

The authors would also like to remind the readers that this work primarily represents the actual experiences that resulted from specific architectural and software choices during the design and implementation of these Grids. The choices made were dictated by the criteria laid out in the first section.

However, the foundation choices of Globus, SRB, and Condor would not be significantly different today than they were four years ago. Nonetheless, if the GGF is successful in its work – and we have every reason to believe that it will be – then in a few years we will see that the functions provided by these packages will be defined in terms of protocols and APIs, and there will be several robust implementations available for each of the basic components.

Some of us (JMB) have experience in building production Grids based on software other than Globus (e.g. UNICORE), the lessons are fundamentally the same as those we have outlined for Globus-based Grids and provide a useful point of reference for confirming the Common Core Functions approach, since both Globus and UNICORE [92] can be viewed as implementations of a subset of the Core Functions and via this can be made to interoperate (e.g. in the Grid Interoperability Project, GRIP [47])"

The impact of the emerging Web Grid Services work is not yet clear. It will likely have a substantial impact on building higher level services, however it is the opinion of the authors OGSI (Grid services [82])will supplement the current Unix runtime system style of Grid environment, and not replace it.

27.6 Acknowledgments

The experience represented in this paper is the result by a lot of hard work by the Grid teams in the author's various organizations.

The principals in the NASA IPG team are Tony Lisotta, Chair, Warren Smith, George Myers, and Judith Utley, all of the NASA Ames Research Center, and Isaac Lopez, of the NASA Glenn Research Center. This project is lead by William Johnston, Tom Hinke, and Arsi Vaziri of NASA Ames Research Center.

The principals in the DOE Science Grid Engineering team are Keith Jackson, Chair, Lawrence Berkeley National Laboratory; Tony Genovese and Mike Helm, ESnet; Von Welch, Argonne National Laboratory; Steve Chan, NERSC; Kasidit Chanchio, Oak Ridge National Laboratory, and; Scott Studham, Pacific Northwest National Laboratory. This project is lead by William E. Johnston, Lawrence Berkeley National Laboratory; Ray Bair, Pacific Northwest National Laboratory; Ian Foster, Argonne National Laboratory; Al Geist, Oak Ridge National Laboratory, and; William Kramer, LBNL / NERSC.

The IPG work is funded by NASA's Aero-Space Enterprise, Computing, Information, and Communication Technologies (CICT) Program (formerly the Information Technology), Computing, Networking, and Information Systems Project. Eugene Tu, Jerry Yan, and Cathy Schulbach are the NASA program managers.

The DOE Science Grid work is funded by the U.S. Dept. of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division (http://www.sc.doe.gov/ascr/mics/) under contract DE-AC03-76SF00098 with the University of California. This program office is lead by Walt Polansky. Mary Anne Scott is the Grids program manager and George Seweryniak is the ESnet program manager.

The architects of UNICORE have done much to identify critical issues in Grids. Dave Snelling and Sven van der Berghe of Fujitsu European Laboratories have developed the influential notion of an Abstract Job Object. Jon MacLaren of the University of Manchester has demonstrated that the EUROGRID CA certificates can interoperate between different Grid middleware systems. This work is funded by the European Union and the German BMBF, the partners in EUROGRID, UNICORE-PLUS and GRIP have all contributed. Dietmar Erwin of FZ Jülich is the project manager and the software integrators are Pallas GmbH under Karl Solchenbach and Hans-Christian Hoppe.

The UK eScience program, funded by the DTI and involving the major UK Science Research Councils, was initiated by John Taylor the Director-General of the Research Councils and is directed on behalf of EPSRC by Tony Hey, on secondment from the University of Southampton.

The European DataGrid work is funded by the European Union and is led by Fabrizio Gagliardi and Bob Jones. The work package leaders are Francesco Prelz, Peter Zoltan Kunszt, Steve Fisher, Olof Barring, John Gordon, François Etienne, Pascale Primet, Frank Harris, Luigi Fusco, Christian Michau, and Maurizio Lancia.

Credit is also due the intellectual leaders of the major software projects that formed the basis of IPG and the DOE Science Grid. The Globus team is led by Ian Foster, University of Chicago and Argonne National Laboratory, and by Carl Kesselman, Univ. of Southern Calif., Information Sciences Institute. The SRB/MCAT team is led by Reagan Moore of the San Diego Supercomputer Center. The Condor project is led by Miron Livny at the Univ. of Wisc., Madison.

27.6 References

[1] **UK eScience Program**, T. Hey, et al. http://www.research-councils.ac.uk/escience/

[2] **European Union DataGrid Project**, EU DataGrid. www.eu-datagrid.org/

[3] **NASA's Information Power Grid**, IPG. http://www.ipg.nasa.gov

[4] **DOE Science Grid**. http://www.doesciencegrid.org

[5] **AP Grid**. http://www.apgrid.org/

[6] **Ninf: A Network based Information Library for Global World-Wide Computing Infrastructure**. http://ninf.apgrid.org/welcome.shtml

[7] **NetCFD: a Ninf CFD component for Global Computing, and its Java applet GUI**, M. Sato, K. Kusano, H. Nakada, S. Sekiguchi and S. Matsuoka. In *Proc. of HPC Asia 2000*. 2000.http://ninf.apgrid.org/papers/hpcasia00msato/HPCAsia2000-netCFD.pdf

[8] **GridLab: A Grid Application Toolkit and Testbed**. http://www.gridlab.org/

[9] **Grid for HPC Applications on Heterogeneous Architectures**, EUROGRID. http://www.eurogrid.org

[10] **The Portable Batch Scheduler**, PBS. http://www.pbspro.com/tech_overview.html

[11] **Cactus**. http://www.cactuscode.org/

[12] **National Energy Research Scientific Computing Center**, NERSC. www.nersc.gov

[13] **European Laboratory for Particle Physics Research**, CERN. http://www.cern.ch

[14] **INFN Grid**, INFN. http://www.infn.it/grid

[15] **The Globus Project**, Globus. http://www.globus.org

[16] **The Condor Project**. http://www.cs.wisc.edu/condor/

[17] **The Storage Resource Broker**. http://www.npaci.edu/DICE/SRB/

[18] **PKI Service - An ESnet White Paper**, T. J. Genovese. September 15, 2000, DOE Energy Sciences Network. http://envisage.es.net/Docs/old%20docs/WhitePaper-PKI.pdf

[19] **ESnet & DOE Science Grid PKI - Overview**. January 24, 2002. http://envisage.es.net/Docs/PKIwhitepaper.pdf

[20] **ESnet's SciDAC PKI & Directory Project - Homepage**, T. Genovese and M. Helm. DOE Energy Sciences Network. http://envisage.es.net/

[21] **Application Experiences with the Globus Toolkit**, S. Brunett, K. Czajkowski, S. Fitzgerald, I. Foster, A. Johnson, C. Kesselman, J. Leigh and S. Tuecke. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*. 1998: IEEE Press.http://www.globus.org/research/papers.html#globus-apps

[22] **Grid Monitoring Architecture Working Group**, Global Grid Forum. http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

[23] **Grid Information Services / MDS**, Globus Project. http://www.globus.org/mds/

[24] **Grid Security Infrastructure (GSI)**, Globus Project. http://www.globus.org/security/

[25] **Globus Resource Allocation Manager (GRAM)**, Globus Project. 2002. http://www-fp.globus.org/gram/overview.html

[26] **The GridFTP Protocol and Software**, Globus Project. 2002. http://www.globus.org/datagrid/gridftp.html

[27] **A Grid Monitoring Architecture**, B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski and M. Swany. http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

[28] **Distributed Monitoring Framework (DMF)**, Lawrence Berkeley National Lab. http://www-didc.lbl.gov/DMF/

[29] **Information and Monitoring Services Architecture**, European Union DataGrid - WP3. http://hepunx.rl.ac.uk/edg/wp3/documentation/doc/arch/index.html

[30] **An XML-Based Protocol for Distributed Event Services.**, W. Smith, D. Gunter and D. Quesnel. In *2001 International Conference on Parallel and Distributed Processing Techniques and Applications*. 2001

[31] **A Framework for Control and Observation in Distributed Environments**, W. Smith. NASA Ames Research Center. http://www.nas.nasa.gov/~wwsmith/papers.html

[32] **Globus I/O**, Globus Project. 2002. http://www-unix.globus.org/api/c-globus-2.0-beta1/globus_io/html/index.html

[33] **Maui Silver Metascheduler**. http://www.supercluster.org/documentation/silver/silveroverview.html

[34] **Condor-G**, J. Frey, T. Tannenbaum, M. Livny, I. Foster and S. Tuecke. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*. 2001: IEEE Press.http://www.globus.org/research/papers.html#Condor-G-HPDC

[35] **Network for Earthquake Engineering Simulation Grid (NEESgrid)**. http://www.neesgrid.org/

[36] **Grid Information Services**, Global Grid Forum. http://www.gridforum.org/1_GIS/gis.htm

[37] **Creating a Hierarchical GIIS**, Globus Project. http://www.globus.org/mds/

[38] ***Understanding X.500 - The Directory***, D. W. Chadwick. 1996. http://www.isi.salford.ac.uk/staff/dwc/X500.htm

[39] **Overview of the Grid Security Infrastructure (GSI)**, Globus Project. 2002. http://www-fp.globus.org/security/overview.html

[40] **DOE Science Grid PKI Certificate Policy And Certification Practice Statement**. http://www.doegrids.org/

[41] **Certification Authorities Acceptance and Feature Matrices**, European Union DataGrid. 2002. http://www.cs.tcd.ie/coghlan/cps-matrix/

[42] **Certification Authorities**, European Union DataGrid. 2002. http://marianne.in2p3.fr/datagrid/ca/ca-table-ca.html

[43] **Design and Deployment of a National-Scale Authentication Infrastructure**, R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer and V. Welch. IEEE Computer, 2000. **33**(12): p. 60-66. http://www.globus.org/research/papers.html#GSI1

[44] **Netscape Certificate Management System**, Netscape. http://wp.netscape.com/cms

[45] **KX.509/KCA**, NSF Middleware Initiative. 2002. http://www.nsf-middleware.org/documentation/KX509KCA/

[46] **ESnet**. http://www.es.net/

[47] **GRid Interoperability Project (Globus adn UNICORE)**, GRIP. http://www.grid-interoperability.org

[48] **Grid Laboratory Universal Environment**, GLUE. http://www.hicb.org/

[49] **NSF Middleware Initiative (NMI)**. http://www.nsf-middleware.org/

[50] **Community Authorization Service (CAS)**, Globus Project. 2002. http://www.globus.org/security/CAS/

[51] **Globus Firewall Requirements**, V. Welch. http://www.globus.org/Security/v2.0/firewalls.html

[52] **Resource Manager for Globus-based Wide-area Cluster Computing**, Y. Tanaka, M. Sato, M. Hirano, H. Nakada and S. Sekiguchi. In *1st IEEE International Workshop on Cluster Computing (IWCC'99)*. 1999.http://ninf.apgrid.org/papers/iwcc99tanaka/IWCC99.pdf

[53] **Performance Evaluation of a Firewall-compliant Globus-based Wide-area Cluster System**, Y. Tanaka, M. Sato, M. Hirano, H. Nakada and S. Sekiguchi. In *9th IEEE International Symposium on High Performance Distributed Computing*. 2000.http://ninf.apgrid.org/papers/hpdc00tanaka/HPDC00.pdf

[54] **The Data Intensive Distributed Computing Research Group**, DIDC. http://www-itg.lbl.gov/DIDC/

[55] **National Laboratory for Applied Network Research**, NLANR. http://dast.nlanr.net/

[56] **NetLogger: A Toolkit for Distributed System Performance Analysis**, D. Gunter, B. Tierney, B. Crowley, M. Holding and J. Lee. In *IEEE Mascots 2000: Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. 2000.http://www-didc.lbl.gov/papers/NetLogger.Mascots.paper.ieee.pdf

[57] **Iperf**, A. Tirumala, F. Qin, J. Dugan, J. Ferguson and K. Gibbs. National Laboratory for Applied Network Research (NLANR). http://dast.nlanr.net/Projects/Iperf/

[58] **Pipechar Network Characterization Service**, G. Jin. http://www-didc.lbl.gov/NCS/

[59] **High-Speed Distributed Data Handling for On-Line Instrumentation Systems**, W. Johnston, W. Greiman, G. Hoo, J. Lee, B. Tierney, C. Tull and D. Olson. In *ACM/IEEE SC97: High Performance Networking and Computing*. 1997.http://www-itg.lbl.gov/~johnston/papers.html

[60] **The NetLogger Methodology for High Performance Distributed Systems Performance Analysis**, B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks and D. Gunter. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*. 1998.http://www-didc.lbl.gov/NetLogger/

[61] **A Network-Aware Distributed Storage Cache for Data Intensive Environments**, B. Tierney, J. Lee, B. Crowley, M. Holding, J. Hylton and F. Drake. In *Proc. 8th IEEE Symp. on High Performance Distributed Computing*. 1999.http://www-didc.lbl.gov/papers/dpss.hpdc99.pdf

[62] **Using NetLogger for Distributed Systems Performance Analysis of the BaBar Data Analysis System**, B. Tierney, D. Gunter, J. Becla, B. Jacobsen and D. Quarrie. In *Proceedings of Computers in High Energy Physics 2000 (CHEP 2000)*. 2000.http://www-didc.lbl.gov/papers/chep.2K.Netlogger.pdf

[63] **Dynamic Monitoring of High-Performance Distributed Applications**, D. Gunter, B. Tierney, K. Jackson, J. Lee and M. Stoufer. In *11th IEEE Symposium on High Performance Distributed Computing, HPDC-11*. 2002.http://www-didc.lbl.gov/papers/HPDC02-HP-monitoring.pdf

[64] **Applied Techniques for High Bandwidth Data Transfers across Wide Area Networks**, J. Lee, D. Gunter, B. Tierney, W. Allock, J. Bester, J. Bresnahan and S. Tuecke. In *Proceedings of Computers in High Energy Physics 2001 (CHEP 2001)*. 2001. Beijing China.http://www-didc.lbl.gov/papers/dpss_and_gridftp.pdf

[65] **TCP tuning Guide for Distributed Applications on Wide Area Networks**, B. Tierney. In *Usenix ;login Journal*. 2001.http://www-didc.lbl.gov/papers/usenix-login.pdf

[66] **A TCP Tuning Daemon**, T. Dunigan, M. Mathis and B. Tierney. In *IEEE Supercomputing 2002*. 2002. Baltimore, Maryland.http://www-didc.lbl.gov/publications.html

[67] **An Online Credential Repository for the Grid: MyProxy**, J. Novotny, S. Tuecke and V. Welch. In *Tenth IEEE International Symposium on High Performance Distributed Computing*. 2001. San Francisco.http://www.globus.org/research/papers.html#MyProxy

[68] **MyProxy**, NCSA. http://www.ncsa.uiuc.edu/Divisions/ACES/MyProxy/

[69] **OpenSSL Certification Authority**, OpenSSL. http://www.openssl.org/docs/apps/ca.html#

[70] **Open-source PKI Book**, 2000. http://ospkibook.sourceforge.net/

[71] **GSI-Enabled OpenSSH**, NCSA. http://www.ncsa.uiuc.edu/Divisions/ACES/GSI/openssh/

[72] **GSI-Enabled FTP**, Globus Project. http://www.globus.org/security/v1.1/index.htm#gsiftp

[73] **Grid-in-a-Box**, Alliance. http://www.ncsa.uiuc.edu/UserInfo/Grid/GiB/

[74] **Grid-in-a-Box Testbed Status**, Alliance. http://www.ncsa.uiuc.edu/UserInfo/Grid/GiB/testbed-status/

[75] **Globus Quick Start Guide**, Globus Project. 2002. http://www.globus.org/toolkit/documentation/QuickStart.pdf

[76] **pyGlobus: a Python interface to the Globus Toolkit**, K. Jackson. Concurrency and Computation: Practice and Experience, 2002. http://www.cogkits.org/papers/c545python-cog-cpe.pdf

[77] **Python Globus (pyGlobus)**, K. Jackson. Lawrence Berkeley National Laboratory. http://www-itg.lbl.gov/gtg/projects/pyGlobus/index.html

[78] **NetSaint**. http://www.netsaint.org/

[79] **Poznan Supercomputing and Networking Center**. http://www.man.poznan.pl/

[80] **Virtual User Account System**, N. Meyer and P. Wolniewicz. http://www.man.poznan.pl/metacomputing/cluster/

[81] **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**, I. Foster, C. Kesselman, J. Nick and S. Tuecke. http://www.globus.org/research/papers.html#OGSA

[82] **Open Grid Service Interface Working Group**, Global Grid Forum. http://www.gridforum.org/ogsi-wg/

[83] **Grid Computing Environments Research Group**, Global Grid Forum. http://www.computingportals.org/

[84] **CoG Kits: Enabling Middleware for Designing Science Appl**, K. Jackson and G. v. Laszewski. Submitted SciDAC proposal, March, 2001, Lawrence Berkeley National Laboratory and Argonne National Laboratory.

[85] **CoG Kits: A Bridge Between Commodity Distributed Computing and High-Performance Grids**, G. v. Laszewski, I. Foster and J. Gawor. In *ACM 2000 Java Grande Conference*. 2000. San Francisco.http://www.globus.org/cog/documentation/papers/index.html

[86] **A Java Commodity Grid Kit**, G. v. Laszewski, I. Foster, J. Gawor and P. Lane. Concurrency: Experience and Practice, 2001. http://www.globus.org/cog/documentation/papers/index.html

[87] **The Grid Portal Development Kit**, J. Novotny. Concurrency - Practice and Experience, 2000. http://www.doesciencegrid.org/Grid/projects/GPDK/gpdkpaper.pdf

[88] **NPSS on NASA's IPG: Using CORBA and Globus to Coordinate Multidisciplinary Aeroscience Applications**, G. Lopez, J. Follen, R. Gutierrez, I. Foster, B. Ginsburg, O. Larsson, S. Martin, S. Tuecke and D. Woodford. 2000.http://www.ipg.nasa.gov/research/papers/NPSS_CAS_paper.html

[89] **A CORBA-based Development Environment for Wrapping and Coupling Legacy Codes**, G. Follen, C. Kim, I. Lopez, J. Sang and S. Townsend. In *Tenth IEEE International Symposium on High Performance Distributed Computing*. 2001. San Francisco.http://cnis.grc.nasa.gov/papers/hpdc-10_corbawrapping.pdf

[90] **LaunchPad**. http://www.ipg.nasa.gov/launchpad/launchpad

[91] **HotPage**. https://hotpage.npaci.edu/

[92] **Uniform Access to Computing Resources**, UNICORE. http://www.unicore.org